

《RESTful Web APIs中文版》

图书基本信息

书名：《RESTful Web APIs中文版》

13位ISBN编号：9787121231158

出版时间：2014-6

作者：[美] Leonard Richardson, Mike Amundsen

页数：382

译者：李哲, 赵震一

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

《RESTful Web APIs中文版》

内容概要

《RESTful Web APIs中文版》是针对RESTful API的实用指南，通过展示各种用来创建高可用应用的强大工具，讲解REST的深层原理，以及介绍基于超媒体API的策略，使读者得以在将上述内容融会贯通后，设计出让客户高度满意的RESTful的web API。《RESTful Web APIs中文版》极具权威性与前瞻性，既代表了API领域的最前沿趋势，也覆盖了API领域的最重要实践。

《RESTful Web APIs中文版》适合所有从事Web开发和架构工作的读者阅读参考。

《RESTful Web APIs中文版》

作者简介

Leonard Richardson, 《Ruby Cookbook》(O'Reilly) 一书的作者, 曾创建了包括Beautiful Soup在内的多个开源代码库。Mike Amundsen 是包括《Building Hypermedia APIs with HTML5 and Node》(O'Reilly) 在内的十几本为人所称道的技术图书的作者。

Sam Ruby 是W3C HTML工作组的联合主席, 同时也是IBM新兴技术组的一名高级技术人员。

书籍目录

序	xix
前言	xxi
第1章 网上冲浪	1
场景1：广告牌	2
资源和表述	2
可寻址性	3
场景2：主页	3
短会话（Short Session）	5
自描述消息（self-descriptive message）	5
场景3：链接	6
标准方法	8
场景4：表单和重定向	9
应用状态（Application State）	11
资源状态（resource state）	12
连通性（connectedness）	13
与众不同的Web	14
Web API 落后于Web	15
语义挑战	16
第2章 一个简单的API	17
HTTP GET：安全的投注	18
如何读取HTTP 响应	19
JSON	20
Collection+JSON	21
向API 写入数据	23
HTTP POST: 资源是如何生成的	24
由约束带来解放	26
应用语义所产生的语义鸿沟	27
第3章 资源和表述	29
万物皆可为资源	30
表述描述资源状态	30
往来穿梭的表述	31
资源有多重表述	32
HTTP 协议语义（Protocol Semantics）	33
GET	35
DELETE	36
幂等性（Idempotence）	36
POST-to-Append	37
PUT	38
PATCH	39
LINK 和UNLINK	40
HEAD	40
OPTIONS	41
Overloaded POST	41
应该使用哪些方法？	42
第4章 超媒体	45
将HTML 作为超媒体格式	46
URI 模板	49

URI vs URL	50
Link 报头	51
超媒体的作用	52
引导请求	52
对响应做出承诺	54
工作流控制	55
当心冒牌的超媒体！	56
语义挑战：我们该怎么做？	57
第5章 领域特定设计	59
Maze+XML：领域特定设计	60
Maze+XML 是如何工作的	61
链接关系	62
访问链接来改变应用状态	64
迷宫集合	65
Maze+XML 是API 吗？	67
客户端1：游戏	68
Maze+XML 服务器	72
客户端2：地图生成器	74
客户端3：吹牛者	76
客户端做自己想要做的事	77
对标准进行扩展	77
地图生成器的缺陷	80
修复（以及修复后的瑕疵）	81
迷宫的暗喻	83
解决语义鸿沟	83
领域特定设计在哪里？	83
最终的奖赏	84
报头中的超媒体	84
抄袭应用语义	84
如果找不到相关的领域特定设计，不要自己制造	86
API 客户端的种类	86
人类驱动的客户终端	86
自动化客户端	87
第6章 集合模式（Collection Pattern）	91
什么是集合？	93
链向子项的集合	93
Collection+JSON	94
子项的表示	95
写入模板（Write Template）	98
搜索模板	99
一个（通用的）集合是如何工作的	100
GET	101
POST-to-Append	101
PUT 和PATCH	101
DELETE	102
分页	102
搜索表单	103
Atom 发布协议（AtomPub）	103
AtomPub 插件标准	105

为什么不是每个人都选择使用AtomPub ?	106
语义挑战：我们应该怎么做?	107
第7章 纯-超媒体设计	111
为什么是HTML?	111
HTML的能力	112
超媒体控件	112
应用语义插件	113
微格式	115
hMaze 微格式	116
微数据	118
改变资源状态	119
为表单添加应用语义	121
与超媒体相对是普通媒体	125
HTML的局限性	126
拯救者HTML5?	127
超文本应用语言	128
Siren	131
语义挑战：我们现在要怎么做?	133
第8章 Profile	135
客户端如何找寻文档?	136
什么是Profile?	137
链接到Profile	137
Profile 链接关系	137
Profile 媒体类型参数	138
特殊用途的超媒体控件	139
Profile 对协议语义的描述	139
Profile 对应用语义的描述	140
链接关系	141
不安全的链接关系	142
语义描述符	142
XMDP：首个机器可读的Profile 格式	143
ALPS	146
ALPS 的优势	150
ALPS 并不是万金油	152
JSON-LD	153
内嵌的文档	156
总结	158
第9章 API 设计流程	161
两个步骤的设计流程	161
七步骤设计流程	162
第1步：罗列语义描述符	163
第2步：画状态图	164
第3步：调整命名	168
第4步：选择一种媒体类型	172
第5步：编写Profile	173
第6步：实现	174
第7步：发布	174
实例：You Type It, We Post It	177
罗列语义描述符	177

画状态图	178
调整名称	179
选择一种媒体类型	180
编写Profile	181
设计建议	182
资源是实现的内部细节	182
不要掉入集合陷阱	183
不要从表述格式着手	184
URL 设计并不重要	184
标准名称优于自定义名称	186
设计媒体类型	187
当你的API 改变时	189
为现有API 添加超媒体	194
改进基于XML 的API	195
值不值得?	196
Alice 的第二次探险	196
场景1：没有意义的表述	196
场景2：Profile	198
Alice 明白了	200
第10章 超媒体动物园	203
领域特定格式	204
Maze+XML	204
OpenSearch	205
问题细节文档	205
SVG	206
VoiceXML	208
集合模式的格式	210
Collection+JSON	211
Atom 发布协议	211
OData	212
纯超媒体格式	219
HTML	219
HAL	220
Link 报头	222
Location 和Content-Location 报头	222
URL 列表	223
JSON 主文档 (Home Documents)	223
Link-Template 报头	224
WADL	225
XLink	226
XForms	227
GeoJSON：一个令人困惑的类型	228
GeoJSON 没有通用的超媒体控件	230
GeoJSON 没有媒体类型	232
从GeoJSON 学习到的经验	233
语义动物园	234
链接关系的IANA 注册表	234
微格式Wiki	235
来自微格式Wiki 的链接关系	236

第11章 API 中的HTTP	241
新HTTP/1.1 规范	242
响应码	242
报头	243
表述选择	243
内容协商 (Content Negotiation)	243
超媒体菜单	244
标准URL (Canonical URL)	245
HTTP 性能	246
缓存 (Caching)	246
条件GET 请求 (Conditional GET)	247
Look-Before-You-Leap 请求	249
压缩	250
部分GET 请求 (Partial GET)	250
Pipelining	251
避免更新丢失问题	252
认证	254
WWW-Authenticate 报头和Authorization 报头	255
Basic 认证	255
OAuth 1.0	256
OAuth 1.0 的缺点	259
OAuth 2.0	260
何时不采用OAuth	261
HTTP 扩展	261
PATCH 方法	262
LINK 和UNLINK 方法	262
WebDAV	263
HTTP 2.0	264
第12章 资源描述和Linked Data	267
RDF	268
RDF 将URL 作为URI 对待.....	270
什么时候使用描述策略	271
资源类型	273
RDF Schema	274
Linked Data 运动	277
JSON-LD	278
将JSON-LD 作为一种表述格式	279
Hydra	280
XRD 家族.....	285
XRD 和JRD	285
Web 主机元数据文档	286
WebFinger	287
本体动物园 (Ontology Zoo)	289
schema.org RDF	289
FOAF	290
vocab.org	290
总结：描述策略生机盎然！	290
第13章 CoAP: 嵌入式系统的REST	293
CoAP 请求	294

《RESTful Web APIs中文版》

CoAP 响应	294
消息种类	295
延迟响应 (Delayed Response)	296
多播消息 (Multicast Message)	296
CoRE Link Format	297
结论：非HTTP 协议的REST	298
附录A 状态法典	301
附录B HTTP 报头法典	325
附录C 为API 设计者准备的Fielding 论文导读	349
词汇表	365

精彩短评

- 1、看一点就有一点的收获。
- 2、总的来说偏工具性质，获得了一些知识，理解更深刻了。但是似乎并没有完全得到想要找的东西。对http和报文重新思考理解；API的话，还需要再温习加深理解，估计要和实践结合效果才好
- 3、刚开始看。个人感觉嘛，前四章貌似泛泛而谈，干货不多。
看到第九章了。感觉翻译的很不好。虽然能读懂，但是中文句子太啰嗦。读起来特别费劲，造成时间上的浪费。
这本书侧重于超媒体API的设计，主要介绍协议，想要学习后端接口RESTFUL设计的同学可以忽略了该书了。
本书侧重于客户端，而不是服务端。
中文翻译实在是太烂了。大部分时间都浪费在理解拗口的中文翻译上了。
- 4、觉得作者应该用一篇Blog就讲完核心的观点，写成了一本书，感觉重点就分散了，读得好痛苦，并且感觉难以应用到现有系统中。唉，服务端毕竟是要受制于客户端的。
- 5、看了一半选择了放弃。。。作者讲的太宏观了
- 6、无意中在图书馆看到，就顺手看完了。虽然是每天都在潜移默化地使用，但毕竟没有看书学习得系统，之前也没能这么透彻的理解缘由还有其中的一些概念.....(好吧，其实我就是没事做在图书馆呆了一晚上随手拿本书看完了
- 7、粗略的讀了一遍，有些内容並不感冒
- 8、难道不应该简简单单就讲完吗。
- 9、太学术了
- 10、去搜索restful的相关博文吧~~
- 11、更多的了解了web api该怎么定义
- 12、估计还要再读一遍
- 13、看了三分之一看不下去了，太无聊了，假装自己看完了吧，嗯。心疼买书钱。追加：一切 url 皆资源，所以不要有动词，嗯。
- 14、有些深奥，没太看懂，略读了下
- 15、作者在不遗余力地推广自己的collection+json，恍惚之间让我回到了XML时代。不过collection+json确实带给了我一些对于REST的别样启发和理解。

《RESTful Web APIs中文版》

精彩书评

1、 本以为会讲什么是restful api,怎么设计好api,有哪些坑,有哪些惯例结果愣是没看明白讲的啥.当初该看看目录再决定买这本书的后悔了...后悔了...后悔了...后悔了...后悔了...后悔了...后悔了...

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com